

Nome: _____ Código: _____

Notas:

- Esta prova é composta por 4 páginas
- Responda às questões de escolha múltipla, indicando a opção correta em maiúsculas
- Cada resposta errada vale -15% da cotação da pergunta

1. Na classe `Info` apresentada a seguir, quais dos membros-função podem alterar o membro-dado `texto`?

```
class Info {  
    string texto;  
public:  
    void f1(string t);  
    void f2(const string &t);  
    void f3(string t) const;  
};
```

- A. Apenas o membro-função `f1`
- B. Apenas os membros-função `f1` e `f2`
- C. Apenas os membros-função `f1` e `f3`
- D. Todos os membros-função `f1`, `f2` e `f3`
- E. Nenhuma das possibilidades anteriores

Resposta: _____

2. É verdade que um membro-função `static`:

- A. Pode usar o apontador `this`
- B. Pode aceder a outros membros (função ou dado) não estáticos
- C. Não pode ser invocado até que a sua classe seja instanciada
- D. Pode ser declarado constante
- E. Nenhuma das possibilidades anteriores

Resposta: _____

3. Na classe `Cbase` o membro-função `virtual void misterio()`

- A. Permite que as classes derivadas de `Cbase` tenham a sua versão do método
- B. Obriga à sua reescrita nas classes derivadas de `Cbase`
- C. Não permite que as classes derivadas de `Cbase` tenham a sua versão do método
- D. Torna a classe `Cbase` abstrata
- E. Nenhuma das possibilidades anteriores

Resposta: _____

4. Considere a classe `Par` parcialmente definida a seguir. Qual das seguintes afirmações é correta?

```
template <class T> class Par {  
protected:  
    T v1, v2;  
public:  
    void iniciar (T v1i, T v2i) { //... }  
    T minimo() { if v1<v2 return v2; else return v1; }  
};
```

- A. A classe `Par`, por ser definida como `template`, necessita de um construtor
- B. A classe `Par`, por ser definida como `template`, não pode ter membros `protected`
- C. O membro-função `minimo` necessita que esteja definido o operador `<` na classe `T`
- D. O membro-função `minimo` não pode retornar um valor do tipo genérico `T`
- E. Nenhuma das possibilidades anteriores

Resposta: _____

5. Considere o seguinte código (incompleto):

```
class Pessoa {  
public:  
    virtual void info() {...}  
};  
  
int main() {  
    Pessoa *p = new Mestrando();  
    p->info();  
}  
  
class Estudante: public Pessoa {  
public:  
    void info() { ... }  
};  
  
class Mestrando: public Estudante {  
public:  
    void info() { ... }  
};
```

A instrução `p->info()` resulta em:

- A. Invocação de `Pessoa::info()`
- B. Invocação de `Estudante::info()`
- C. Invocação de `Mestrando::info()`
- D. O programa está errado, pois o método `virtual void info()` não pode ter implementação
- E. Nenhuma das possibilidades anteriores

Resposta: _____

6. A inserção de um elemento num vetor ordenado (mantendo a ordenação) é realizada pelo seguinte algoritmo: uso de pesquisa binária para localizar a posição onde deve ser inserido o novo elemento; inserir o novo elemento nessa posição. Qual a complexidade temporal deste algoritmo?

- A. $O(\log N)$
- B. $O(N \log N)$
- C. $O(N^2)$
- D. $O(N)$
- E. Nenhuma das possibilidades anteriores

Resposta: _____

Nome: _____ Código: _____

7. Indique a complexidade temporal da seguinte função:

```
int funcaoQQ(vector<int> &v) {  
    int ct=0;  
    for(int i=0; i<10; i++) {  
        for(int j=v.size()-1; j>i; j/=2)  
            if (v[i]<v[j]) ct++;  
    }  
    return ct;  
}
```

- A. $O(\log n)$
- B. $O(n)$
- C. $O(n \log n)$
- D. $O(n^2)$
- E. Nenhuma das possibilidades anteriores

Resposta: _____

8. Defina o conceito de classe abstrata. Qual a utilidade de uma classe abstrata?

9. Enumere duas vantagens de, numa função que inclui como argumento um objeto da classe x , este ser passado por referência constante e não por valor.

10. O algoritmo de ordenação Quicksort possui complexidade temporal $O(n \log n)$, no caso médio, mas pode, no pior caso, apresentar complexidade temporal $O(n^2)$. Explique porquê.